基于对抗神经网络的恶意用户检测

第39组 蔡雨凡 闫璐

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Task Definition

Malicious detection

Recently, a new type of attack, coined Sybil Account comes out targeting at social networks. Sybil are accounts that post fake reviews to a certain store or service in campaigns and get paid.

Node classification

Two type of information: the feature of the node and the network.

Graph Representation Learning

Graph representation learning tries to embed each node of a graph into a low-dimensional vector space, which preserves the structural similarities or distances among the nodes in the original graph.

Twitter:

User-Network

This dataset consists of 81,306 nodes representing users and, 1,768,149 edges representing relationships.

http://snap.stanford.edu/data/ego-Twitter.html

User-Labels

This dataset is achieved from a paper called POISED: Spotting Twitter Spam Off the Beaten Paths. The tweets in this dataset were manually checked by a group of 14 security researchers who labeled them independently.

dianping: The dataset was crawled on Dianping from January 1, 2014 to June 15, 2015 and includes 10,541,931 reviews, 32,940 stores, and 3,555,154 users.

Cora: The dataset consist of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. Each pubilication in the dataset is described by a 0/1-valued work vector indicating the absence/presence of the corresponding word from the dictionary. The dictionary consists of 1433 unique words.

GNN

Convolution Neural Network – extract!

1. Discrete convolution in CNN: filter for shared parameters
2. Convolution operation in CNN: [feature map] by calculating the central pixel point and the [weighted sum] of adjacent pixel points;

Reasons for studying GCN

1. CNN's [translation invariance] is not applicable on [non-matrix structure] data.
2. Hope to extract spatial features on the [topology map] for machine learning

Two ways to extract the spatial features of [topology]

1. vertex domain (spatial domain):

operation: find the neighbors adjacent to each vertex, and use feature representation

Examples: GraphSage

2. spectrum domain:

Spectral domain process:

(1) Define the Fourier Transformation Fourier transform on the graph (using Spectral graph theory, study the properties of the graph by means of the eigenvalues and eigenvectors of the **Laplacian matrix of the graph**)

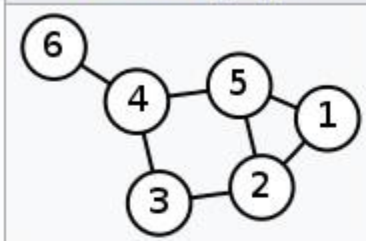
(2) Define the convolution on the graph convolution

Examples: GCN

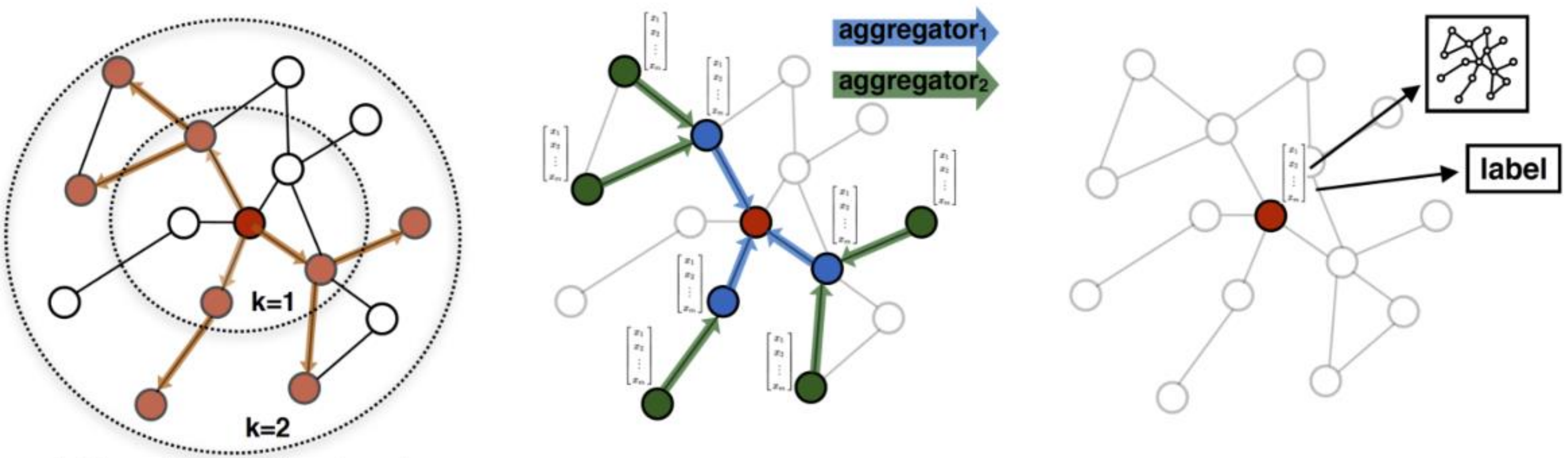
Laplacian matrix of the graph：

$$L = D - A$$

Where L is the Laplacian matrix and D is the degree matrix of the vertex (diagonal matrix), the elements on the diagonal are sequentially the degrees of the respective vertices, and A is the adjacency matrix of the graph.

Labeled graph	Degree matrix	Adjacency matrix	Laplacian matrix
	$\begin{pmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 3 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$	$\begin{pmatrix} 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$	$\begin{pmatrix} 2 & -1 & 0 & 0 & -1 & 0 \\ -1 & 3 & -1 & 0 & -1 & 0 \\ 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & -1 \\ -1 & -1 & 0 & -1 & 3 & 0 \\ 0 & 0 & 0 & -1 & 0 & 1 \end{pmatrix}$

GraphSage



1. Sample neighborhood

2. Aggregate feature information from neighbors

3. Predict graph context and label using aggregated information

GAN

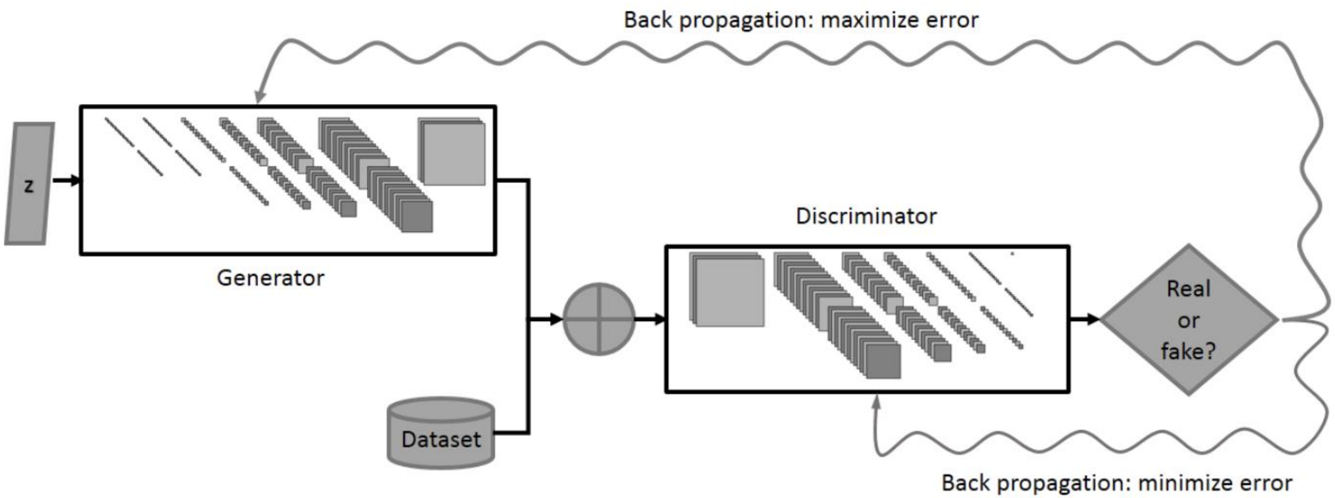


Fig 1.0, Source : Nvidia.

$$\min_G \max_D V(D, G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))].$$

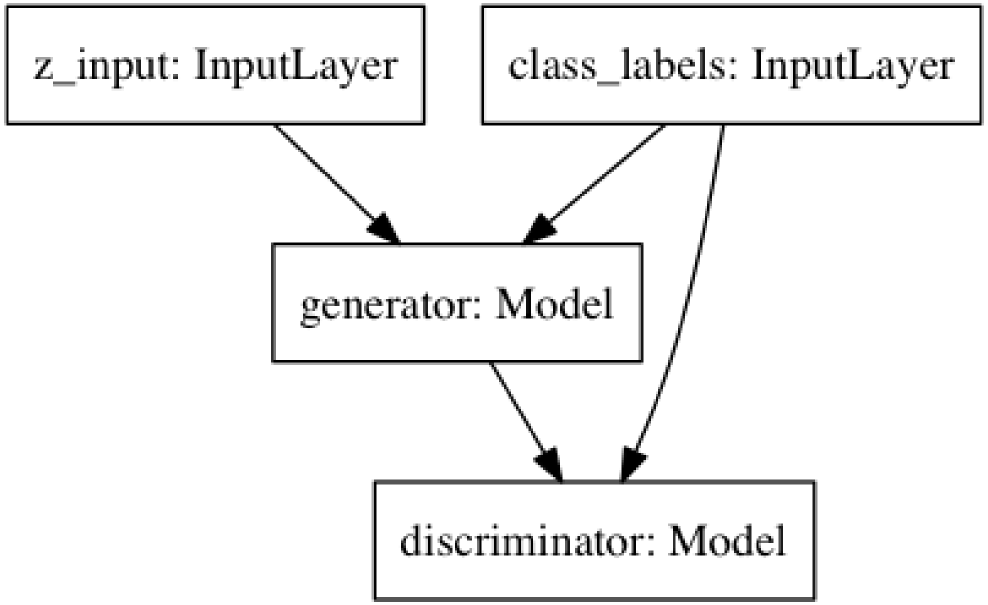
Discriminator Loss:

$$\frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)\right]$$

Generator Loss:

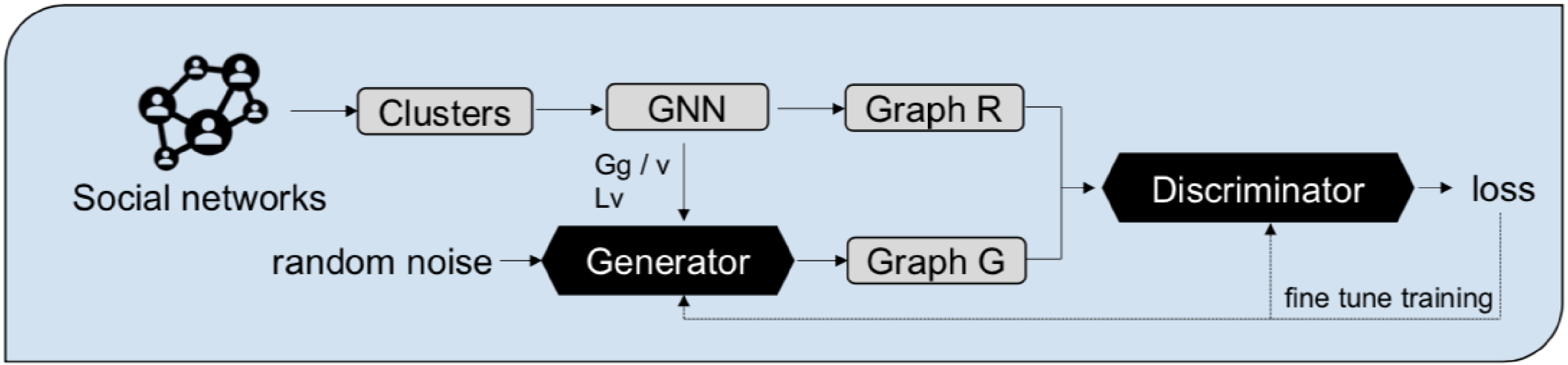
$$\frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right)$$

GAN



Conditional-GAN: https://arxiv.org/abs/1411.1784
Use random noise and real data with embedded labels as input

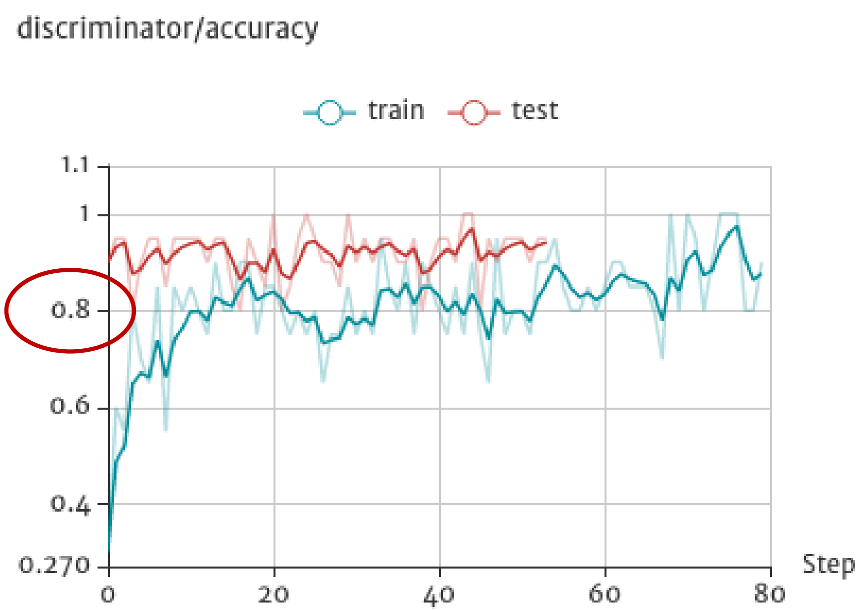
GAN with GNN result



Random noise replaced by neighbor nodes’ features predicted by GNN

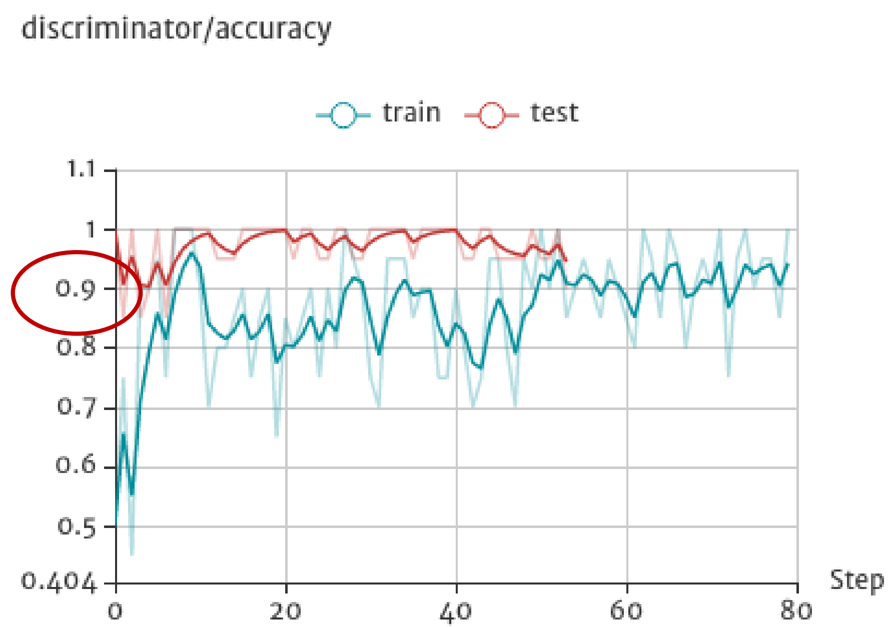
GAN

--training with random noise



Average test accuracy:0.914815

--training with GNN result



Average test accuracy:0.972222

Results of Sybil dataset

	Decision Tree	SVM	GNB	KNN	Adaboost	Random Forest	GCN/ GraphSage (no feature)	GAN (random)	GAN (+GCN)
Loss							0.344	0.109	0.173
Accuracy	0.659	0.677	0.875	0.595	0.785	0.897	0.80/ 0.81	0.913	0.963
Precision	0.828	0.716	0.832	0.817	0.817	0.847	0.811	0.951	0.962
recall	0.655	0.698	0.667	0.577	0.567	0.620	0.577	0.971	0.975

Results of Cora dataset

	Decision Tree	SVM	GNB	KNN	Adaboost	Random Forest	GCN	GAN (random)	GAN (+GCN)
loss							0.7207	0.458	0.283
Accuracy	0.618	0.647	0.484	0.427	0.557	0.664	0.8340	0.918	0.972
Precision	0.626	0.710	0.486	0.440	0.597	0.673	0.740	0.765	0.835
recall	0.626	0.646	0.484	0.427	0.557	0.657	0.811	0.972	0.976

Analysis

Thanks!